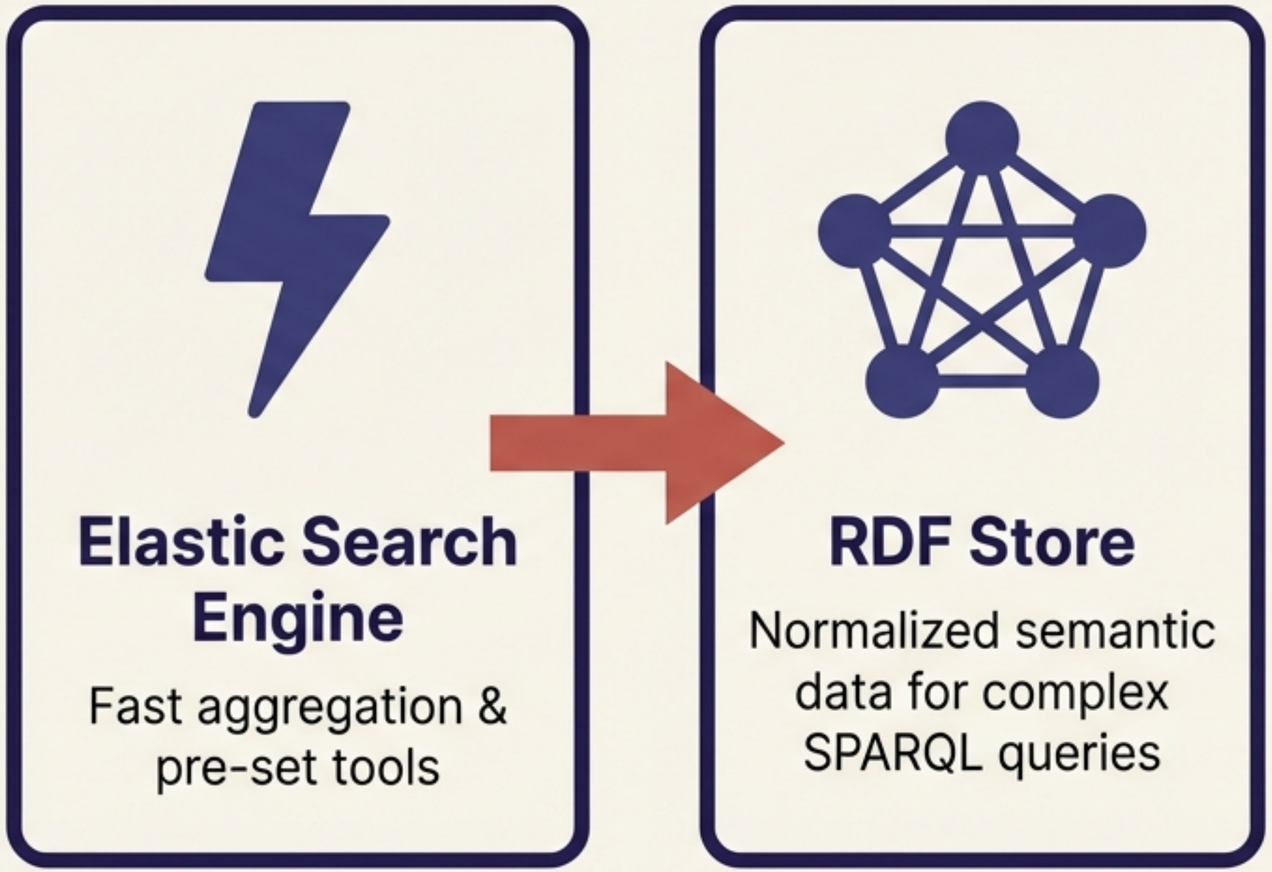
Japan Search RDF Model Primer

A Visual Guide to Querying Cultural Heritage Data

NotebookLM

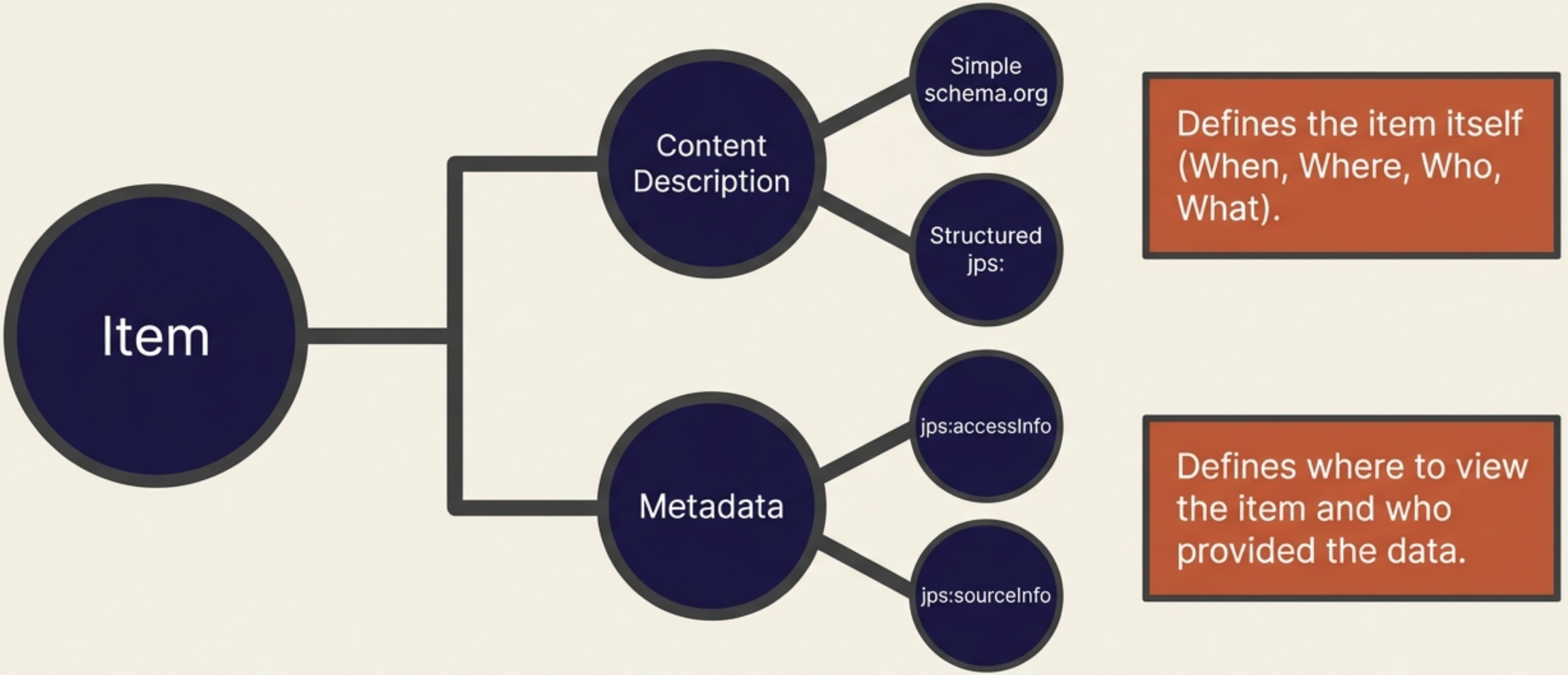
Japan Search pairs an elastic search engine with an RDF store for deep normalization.

- JPS aggregates millions of cultural artifacts—from national museum collections to broadcast programs and anime.
- Disparate source data is heavily normalized into a highly structured RDF Semantic Web graph to allow precise, cross-sector discovery.



NotebookLM

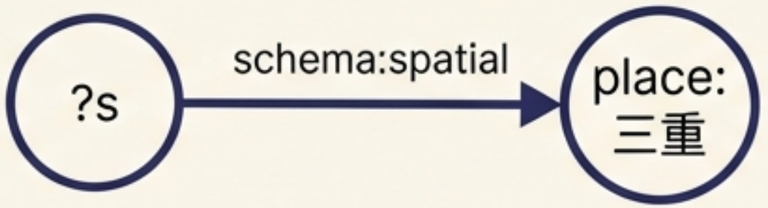
The graph separates normalized content descriptions from access and source metadata.



NotebookLM

Simple schema properties provide broad matches, while structured descriptions define precise relationships.

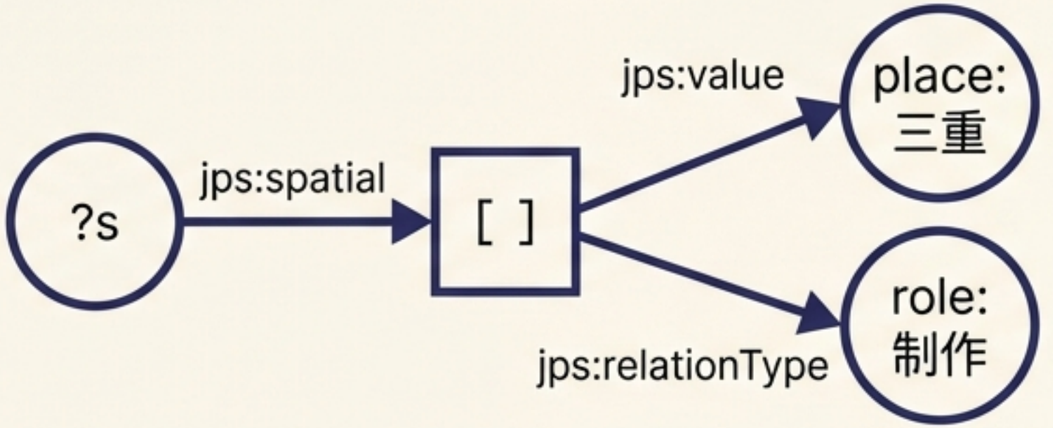
Simple




```
?s schema:spatial place:三重 .
```

(Retrieves anything related to Mie)

Structured

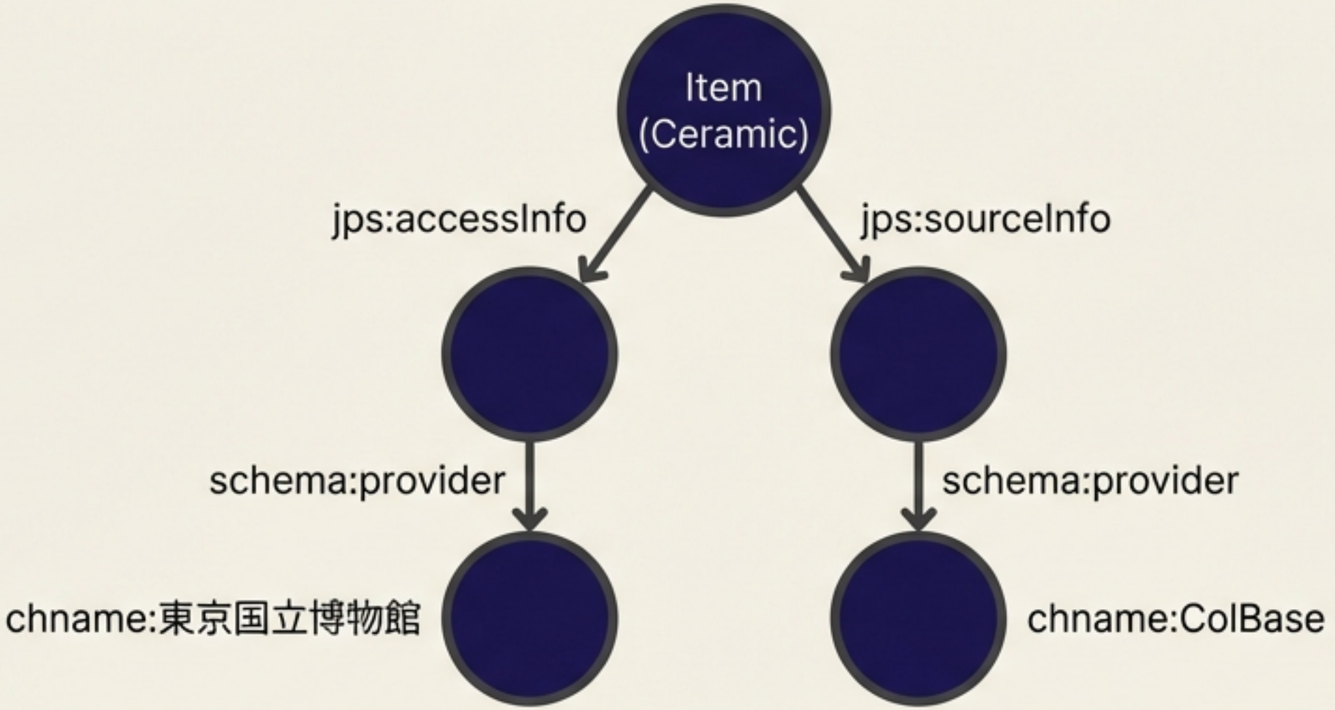



```
?s jps:spatial [
  jps:value place:三重 ;
  jps:relationType role:制作 ] .
```

(Retrieves only items created in Mie)

NotebookLM

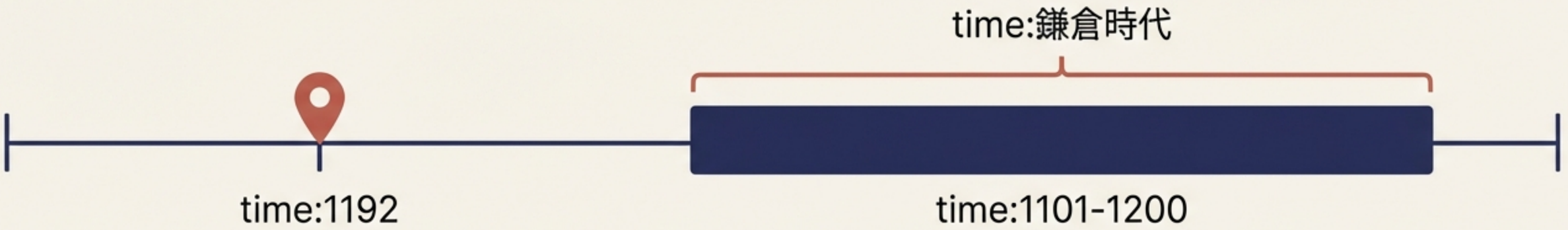
Access and source metadata trace an item’s physical location and its digital provider.



- Use **jps:accessInfo** to find where an item is exhibited or hosted.
- Use **jps:sourceInfo** to isolate the specific database (e.g., ColBase) that supplied the metadata to Japan Search.

NotebookLM

Temporal queries capture exact years, distinct eras, and overlapping time intervals.




```
?when jps:start ?start ;
      jps:end ?end .
FILTER (?start > 1100 && ?end <= 1200)
```

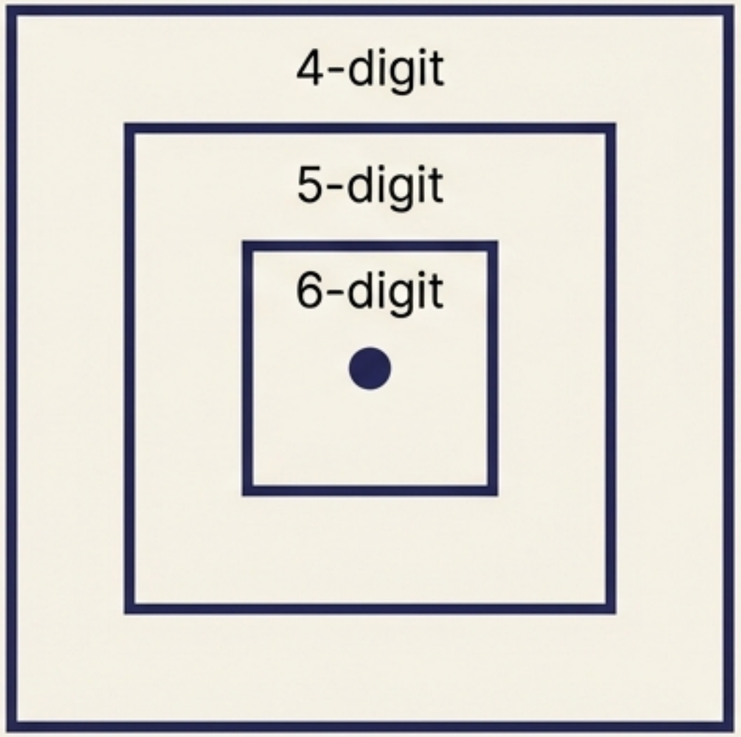
```
?s jps:temporal [ jps:era time:鎌倉時代 ] .
```

Note: All time resources in JPS (even exact years) contain both a `jps:start` and `jps:end` value to allow interval math.

NotebookLM

Geohash URIs and property paths enable spatial radius searches

Locations are normalized to prefectures (**place**:), but detailed items include exact **schema:geo** lat/longs and calculated 4-to-6 digit Geohash URIs.



The SPARQL 1.1 property path operator (? or +) traverses `jps:within` to automatically retrieve items in surrounding, wider geographic grids.

```
?s jps:spatial/schema:geoCoveredBy/jps:within?
  http://geohash.org/xn1pq
```

NotebookLM

Linked Open Data solves the inconsistencies of Latin transcription.

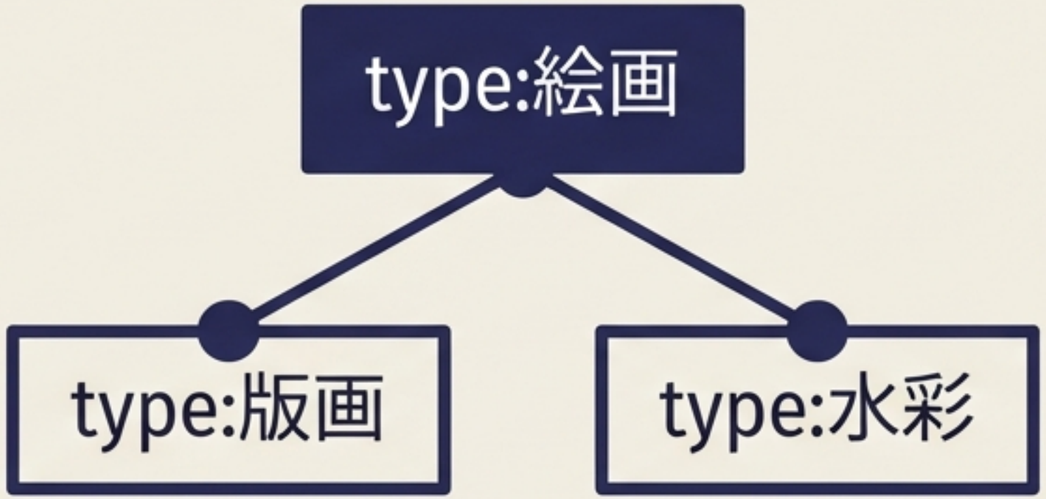
Relying on string matching for translated names fails due to wild spelling variations. JPS normalizes creators into the chname: namespace.

```
?s jps:agential [
  jps:value/owl:sameAs* chname:夏目漱石 ] .
```

The zero-or-more path operator (*) catches the exact JPS name, or routes through **external authority IDs** seamlessly.

NotebookLM

Class hierarchies automatically capture specialized subtypes in broad categorical queries.




```
?type rdfs:subClassOf* type:絵画
```

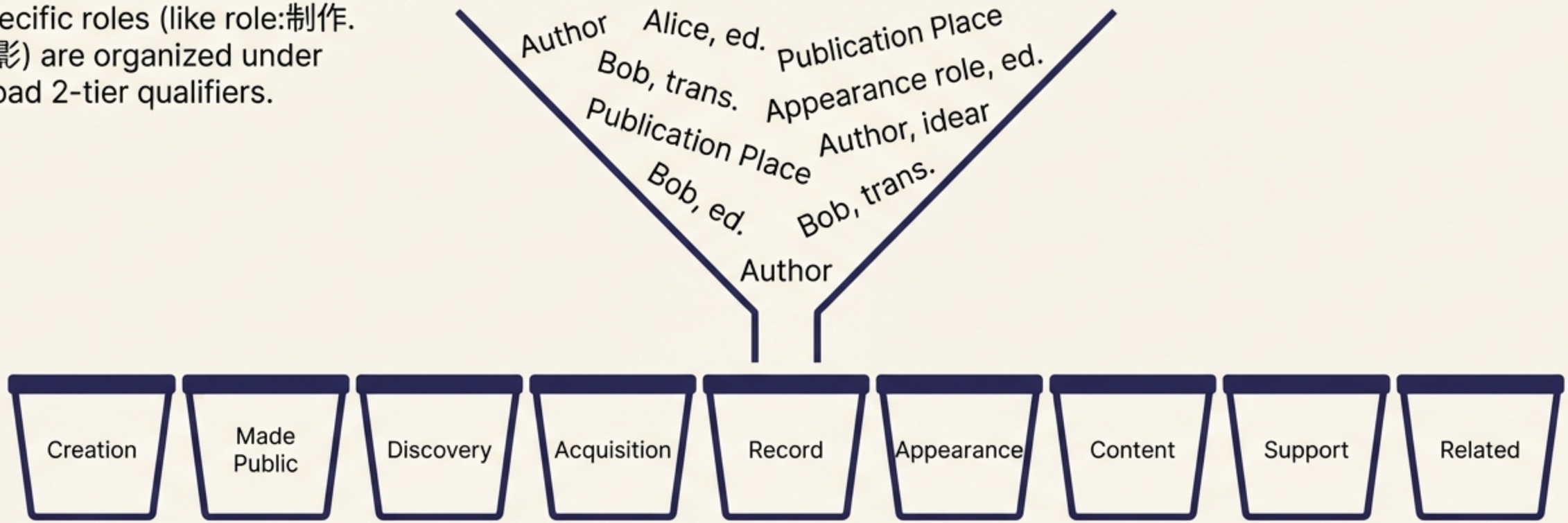
Data providers classify items differently (e.g., Hokusai as ‘Painting’ vs ‘Print’). The subclass operator catches all downstream variants.

Subject Queries: Use schema:about combined with JPS keywords (keyword:食) or external classifications (NDLSH, NDC).

NotebookLM

Nine basic role groups simplify queries across wildly varying source terminologies.

Specific roles (like role:制作. 撮影) are organized under broad 2-tier qualifiers.




```
?role skos:broader? role:公開
```

Adding **skos:broader?** instantly captures all highly specific sub-roles categorized under a basic group like ‘Made Public’.

NotebookLM

Deep linkages surface high-resolution IIIF manifests and normalize fragmented license versions.

Query `schema:image` to retrieve direct thumbnail URLs.

Check the access URL type:

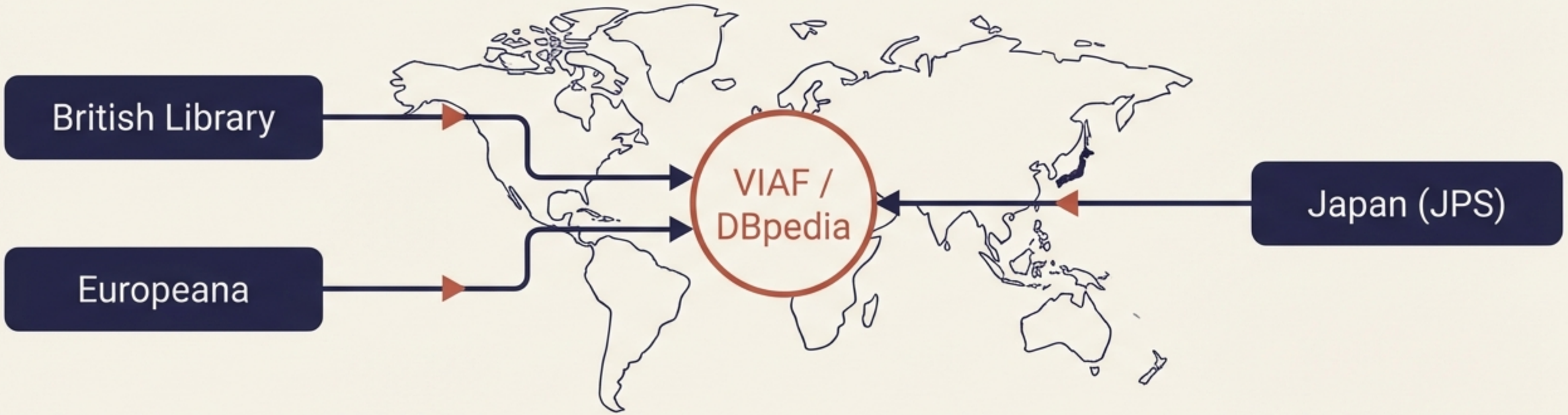
```
?manifest a http://iiif.io/api/presentation/2#Manifest
```

Variations in CC licenses (like v4.0 vs v3.0) are normalized. Query `dct:isVersionOf` to catch all compatible variants of a license like CC BY-SA.

NotebookLM

Federated queries seamlessly merge Japan Search graphs with the British Library and Europeana.

Query multiple global databases in a single SPARQL script using the SERVICE keyword.

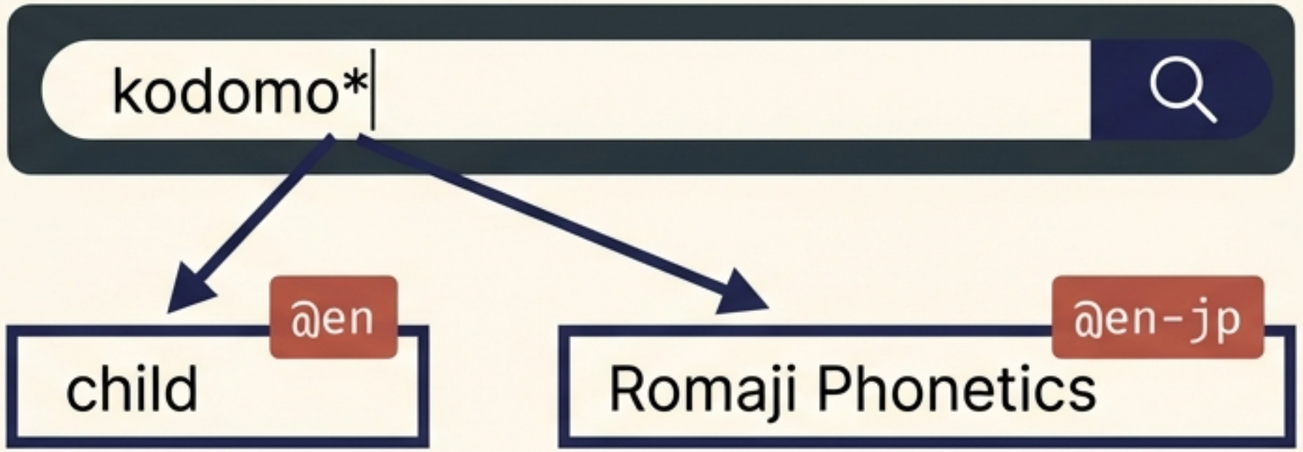


Extract the standard LOD URI using rdfs:isDefinedBy, pass it to the external endpoint, and UNION the results.

```
SERVICE http://bnb.data.bl.uk/sparql { ?creator owl:sameAs ?viaf ... }
```

NotebookLM

Language tags and full-text indexing bypass Japanese transcription barriers



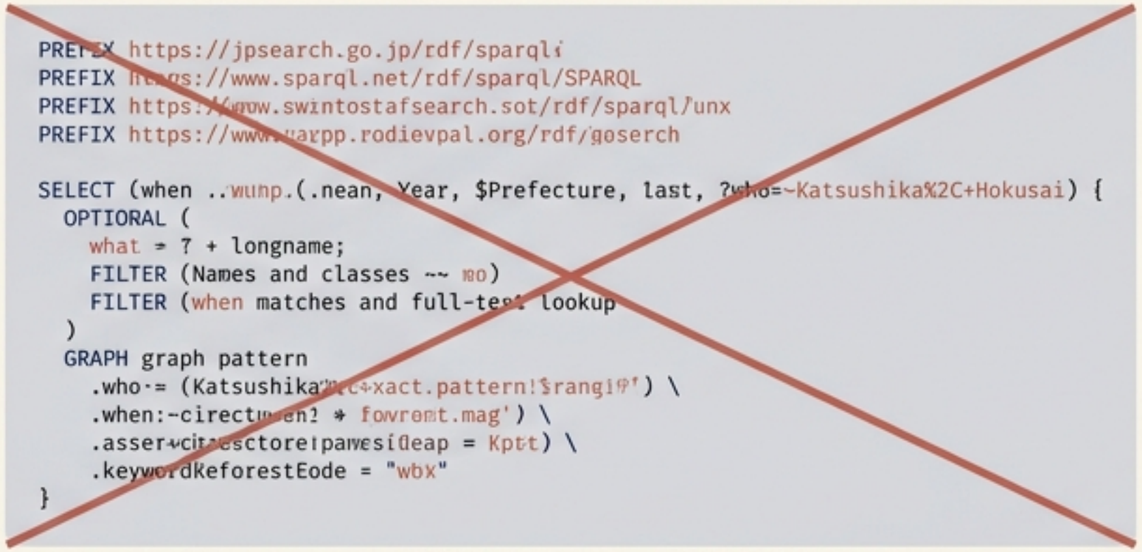
- Entities include schema:name values tagged for language. Use “@en” for exact English labels.
- JPS generates machine-read Japanese romaji tagged as “@en-JP” to allow broad phonetic querying.
- Use Virtuoso’s proprietary full-text function to execute heavy text searches without timing out.

```
FILTER bif:contains(?name, '"kodomo*""')
```

NotebookLM

The EasySPARQL API translates simple REST parameters into complex graph queries.

Build lightweight applications by passing basic parameters. The server constructs the SPARQL and returns standard JSON or XML.



```
PREFIX https://jpsearch.go.jp/rdf/sparql/
PREFIX https://www.sparql.net/rdf/sparql/SPARQL
PREFIX https://www.swintostafsearch.sot/rdf/sparql/unx
PREFIX https://www.warpp.rodievpal.org/rdf/goserch

SELECT (when ..wump.(.nean, Year, $Prefecture, last, ?who=~Katsushika%2C+Hokusai) {
  OPTIORAL (
    what = ? + longname;
    FILTER (Names and classes ~~ mo)
    FILTER (when matches and full-tex? lookup
  )
  GRAPH graph pattern
    .who·= (Katsushika"rc4xact.pattern!$rangi?') \
    .when:~circumen? * fovrom.mag') \
    .asser+citesctore!panesíleap = Kptt) \
    .keywordkeforestEode = "wbx"
}
```

https://jpsearch.go.jp/rdf/sparql/easy/?who=~Katsushika%2C+Hokusai

Parameter	Value
when	Year (YYYY) or Japanese Era name
where	Prefecture or exact lat,long
who / what	Names and classes (use ~ prefix to auto-include aliases/subclasses)
keyword / text	Subject matches and full-text lookup

NotebookLM

Quick Reference: Essential Japan Search Namespaces.

`schema:`	Simple properties (e.g., spatial, temporal, creator)
`jps:`	Structured graph properties (e.g., agential, relationType, accessInfo)
`chname:`	Normalized cultural heritage entity names (People, Organizations)
`time:`	Normalized temporal nodes (Years, Eras)
`place:`	Normalized locations (Prefectures)
`type:`	Core item classification (Painting, Ceramic, Book)
`role:`	Standardized 9-group action classifications

NotebookLM